Spatial-temporal City-scale Congestion Prediction using a two-stream Graph Neural Network

Ye Wei
Department of Computer Science
Loughborough University
Loughborough, UK
Y.Wei@lboro.ac.uk

He Haitao
School of Architecture, Building and Civil Engineering
Loughborough University
Loughborough, UK
H.He@lboro.ac.uk

Hui Fang
Department of Computer Science
Loughborough University
Loughborough, UK
H.Fang@lboro.ac.uk

October 31, 2022

Abstract

Accurate short-term city-scale traffic congestion forecasting can improve traffic operation and control. To this aim, the application of graph neural networks (GNNs) can potentially increase prediction accuracy, but two major challenges exist. These include the homophilous predictions and heavy computational burden for large-scale congestion forecasting, which deteriorate the generalisability of current models and limit their performance. To address these challenges, in this short report, we propose a two-stream GNN architecture to tackle these issues. Training with the Traffic4cast competition 2022 data, i.e. a traffic dataset collected via loop counters from three capital cities, including London, Madrid and Melbourne, we extract city topology representations and temporal features from each loop counter node respectively before we concatenate them for the final congestion prediction. This design significantly reduces the complexity of our GNN architecture and enhances the non-homophilous prediction by decoupling global topology and local temporal features before exploring their correlations. The results demonstrate that a light and generalisable model can be built and well trained efficiently. All source code is available at `https://github.com/Ye-We1/Traffic4cast2022`.

1 Introduction

Traffic forecasting problem, especially short-term traffic congestion prediction, has been a significant topic in transportation research for more than two decades [9]. By providing accurate prediction, potential congestion can be alleviated, and individual drivers' driving experience can be enhanced. However, the complex spatial-temporal dependencies among traffic flows still hamper current research in traffic forecasting[4]. Different from other spatial-temporal learning tasks, such as Video Semantic Segmentation[2], the lack of a large-scale benchmark dataset hinders the analysis of spatial-temporal features by data-driven methods.

Aiming to facilitate a fair comparison between different methods, the Institute of Advance Research in Artificial Intelligence (IARAI) has hosted a competition series called Traffic4cast since 2019 [8]. Participants are encouraged to apply existing spatial-temporal prediction algorithms to real-world datasets collected by HERE Technologies[3]. In the past three competitions from 2019 to 2021, various models have been tested on their ability to predict traffic speed and volume[6, 5]. U-net, an encoder-decoder model designed initially for medical image segmentation, is shown to be one of the most competitive frameworks [1]. Unlike the converted image data to represent traffic flows in previous

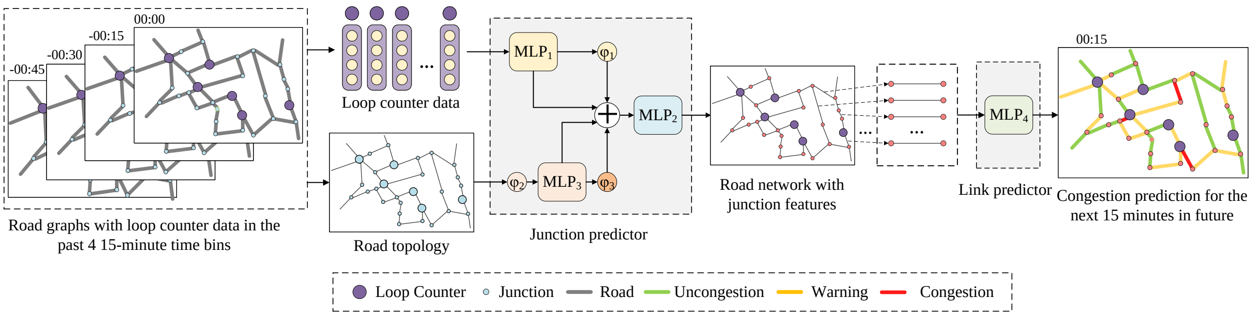

Figure 1: The overview of the proposed two-stream GNN architecture.

competitions, the data collected for the Traffic4cast 2022 is in a form of graphs (i.e., nodes features and link labels), as it is a natural way to represent traffic flows via loop counters which are sparsely set across the entire city [8]. Intuitively, GNN is one of the most suitable neural network architecture to process the data.

There are two bottlenecks to limit the application of GNN to road congestion prediction problem: (i) since the learning objective of GNN is minimising the overall prediction errors, graph learning tends to make the prediction of neighbour nodes close to each other. However, this over-smoothness assumption could lower the performance on a non-homophily graph, i.e., the traffic graph in this case; and (ii) the scalability issue when dealing with the graphs representing large-scale cities. Typically, a powerful GPU server is demanded to achieve good performance for city-scale congestion prediction.

In this paper, inspired by [7], we design a two-stream GNN architecture to tackle the above-mentioned challenges. Specifically, we decouple representations of road topology and traffic temporal states at each junction into two branches via two Multilayer Perceptron (MLP) networks. After using two linear projectors to further enhance these representations, they are combined to produce node features with embedded topology features. Subsequently, by pairing between starting and ending nodes, these features are reshaped to feed into another MLP for the road congestion prediction. We present the details of our proposed methods in the following section.

2 Proposed Method

2.1 Network architecture

As depicted in Figure 1, we deploy two MLPs to extract features from loop counter data and road topology data via two separate streams before we combine them and use another MLP to further extract the node features. The node feature extraction process can be expressed as the following equations:

$$\begin{aligned}\mathbb{H}_\mathrm{N} &= \mathrm{MLP}_1(\mathbb{N})\\ \mathbb{H}_\mathrm{A} &= \mathrm{MLP}_3(\varphi_2\mathbb{A})\\ \mathbb{H} &= \sigma\mathrm{MLP}_2(\sigma(\mathbb{H}_\mathrm{N} + \mathbb{H}_\mathrm{A} + \varphi_1\mathbb{H}_\mathrm{N} + \varphi_3\mathbb{H}_\mathrm{A}))\end{aligned}$$

where $\mathbb{N} \in \mathbb{R}^{n,4}$ is the loop counter data, and n is the number of junctions. $\mathbb{A} \in \mathbb{R}^{n,n}$ is the adjacency matrix of the road network. $\varphi_2 \in \mathbb{R}^{h,n}$ is a trainable weight matrix and h is the number of hidden channels. Similar to φ_2, $\varphi_1 \in \mathbb{R}^{h,h}$ and $\varphi_3 \in \mathbb{R}^{h,h}$ are two trainable weight matrices. With this structure, the non-homophily patterns are preserved better since the two-stream design enhances the loose-coupling characteristics between node features and their topology constraint.

Subsequently, these node features are paired by identifying the starting and ending nodes of each road. We deploy another two-layer MLP to get the congestion prediction of each road as the final output.

2.2 Training

In the training process, the input data of our model is graphs in which the nodes represent junctions and the edges represent roads. If a junction is equipped with a loop counter, its corresponding node has four measurements representing the values of loop counter data in the past four 15-minute temporal bins. Otherwise, the four measurements are NaN. Similarly, the edges are classified into one of three classes, including uncongestion, warning and congestion, if data is available, and annotated as NaN with missing labels. Before putting the graph data into the model, we set all NaN values to -1. The loss function we used is the cross entropy loss, with ignore index = -1 when processing NaN inputs or outputs.

3 Experiment results

3.1 Data and training settings

The dataset of the Traffic4cast 2022 competition consists of data collected from three cities: London, Madrid and Melbourne. Like the previous competitions, the data is available for download from Here Technologies. For the training and validation of our model, we use two categories of data: traffic loop counter data as the input data and congestion classes as the ground truth labels. The traffic loop counter data is the new input data for the Traffic4cast 2022. Although they are spatially sparse, they capture traffic flows at their locations. For the three competition cities, the loop counter measurements are represented as nodes with measured volumes at every 15-minute temporal intervals. These data are then attached to the nodes of each road graph. Congestion class data contains the labels for each segment of the road graph: red / congestion, yellow / warning, and green / uncongestion. The class is derived from the aggregated GPS probe data. If data are not enough to derive the congestion class, it will output missing values. In the competition, the congestion data is also presented in the form of graphs.

At training time, the loop counter data in the past four 15-minute time bins are used as the input node features, while the congestion data in the future 15-minute interval is used as the output edge labels. When training our model, we used all the provided data since it achieved the best performance on the hidden test data. The size of each city's graph is shown in table.1. For the training samples, we can generate $(24 \times 4) - 4 = 92$ samples for each day since 24-hour data per day is provided. Consequently, we have 9752 training samples for Melbourne, 10120 training samples for London and 10028 training samples for Madrid respectively.

Table 1: Training Graph Size

Graph size			
Nodes	Edges	City	Period
49510	94871	Melbourne	2020-06-01 - 2020-12-30
63397	121902	Madrid	2021-06-01 - 2021-12-31
59110	132414	London	2019-07-01 - 2020-01-31

We construct models for individual cities and train them with their corresponding training data. Each model is trained for 100 epochs and each epoch having 9752 samples for Madrid, 10120 samples for London and 10028 samples for Madrid respectively. All models are trained on a single NVIDIA GeForce RTX 2080 Ti GPU and the batch size is set to 1. Following the baseline model provided by the organizer, a masked cross-entropy loss function is optimised with the Adam Optimiser. The training setting details can be found in Table 3.

3.2 Final performance

Our model ranked at the 8_{th} place in the Traffic4cast core competition, and the top ten teams and their results are listed in table.2. Although our model did not achieve top performance, its two advantages has two-folds: (1) our model is light-weight and can be trained efficiently. By using a single RTX 2080Ti GPU card with limited memory, we still reach a very competitive performance, which is 0.034 lower than the top model; and (2) we believe our model has strong generalisability since the network is purposely designed for non-homophilic graph. We will further evaluate our model on cities with more inconsistent traffic flows.

Table 2: Core competition leaderboard results for the top 10 teams

Rank	Team	Cross entropy loss
1	ustc-gobbler	0.8431
2	Bolt	0.8496
3	oahciy	0.8504
4	GongLab	0.8560
5	AP_DE	0.8735
6	TSE	0.8736
7	discovery	0.8759
8	**ywei**	**0.8778**
9	IARAI_DS	0.8913
10	STIL@home	0.9305

Table 3: Training details

model	MLP-based GNN
fine-tuning/freezing	/
sampling	all data
size of training set	106*92 for Melbourne 110*92 for London and 109*92 for Madrid
number of iterations	100 epochs
batch size	1
trainable parameters/model size	526.1K
optimizer and schedule	AdamW

4 Conclusions and discussions

In this paper, we present a simple yet effective framework for congestion prediction in real world. Experimental results demonstrate that our approach beats the competition baseline model and is comparable to many more complex models.

Since our aim was to find an efficient way to make congestion predictions, we did not spend much time improving the model structure. Applying attention-based layers or other techniques may help our model to achieve better results. Using other tricks, such as the ensemble model, may also give better results.

References

[1] Christian Eichenberger et al. "Traffic4cast at NeurIPS 2021 - Temporal and Spatial Few-Shot Transfer Learning in Gridded Geo-Spatial Processes". In: *Proceedings of the NeurIPS 2021 Competitions and Demonstrations Track*. Vol. 176. Proceedings of Machine Learning Research. PMLR, Dec. 2022, pp. 97–112. URL: `https://proceedings.mlr.press/v176/eichenberger22a.html`.

[2] Alberto Garcia-Garcia et al. "A survey on deep learning techniques for image and video semantic segmentation". In: *Applied Soft Computing* 70 (2018), pp. 41–65. DOI: `https://doi.org/10.1016/j.asoc.2018.05.018`.

[3] *Here Technology*. URL: `https://www.here.com/`.

[4] Weiwei Jiang and Jiayun Luo. "Graph neural network for traffic forecasting: A survey". In: *Expert Systems with Applications* 207 (Nov. 2022), p. 117921. DOI: `10.1016/j.eswa.2022.117921`.

[5] Michael Kopp et al. "Traffic4cast at NeurIPS 2020 - yet more on the unreasonable effectiveness of gridded geo-spatial processes". In: *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*. Vol. 133. Proceedings of Machine Learning Research. PMLR, Dec. 2021, pp. 325–343. URL: `https://proceedings.mlr.press/v133/kopp21a.html`.

[6] David P Kreil et al. "The surprising efficiency of framing geo-spatial time series forecasting as a video prediction task – Insights from the IARAI t4c Competition at NeurIPS 2019". In: *Proceedings of the NeurIPS 2019 Competition and Demonstration Track*. Vol. 123. Proceedings of Machine Learning Research. PMLR, Dec. 2020, pp. 232–241. URL: `https://proceedings.mlr.press/v123/kreil20a.html`.

[7] Derek Lim et al. "Large Scale Learning on Non-Homophilous Graphs: New Benchmarks and Strong Simple Methods". In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021, pp. 20887–20902.

[8] *Traffic4cast 2022*. URL: https://www.iarai.ac.at/traffic4cast/.

[9] Eleni I. Vlahogianni, Matthew G. Karlaftis, and John C. Golias. "Short-term traffic forecasting: Where we are and where we're going". In: *Transportation Research Part C: Emerging Technologies* 43 (2014), pp. 3–19. DOI: https://doi.org/10.1016/j.trc.2014.01.005.